IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHER DISTRICT OF OHIO

NORSYNC TECHNOLOGY AS,
 	
            Plaintiff,
 	
    v.
 	
ORACLE CORPORATION,
 	
            Defendant.
	Civil Action No.:
	
	TRIAL BY JURY DEMANDED

COMPLAINT FOR INFRINGEMENT OF PATENT

Now comes Plaintiff, Norsync Technology AS ("Plaintiff" or "Norsync"), by and through undersigned counsel, and respectfully alleges, states, and prays as follows:

NATURE OF THE ACTION

1.   This is an action for patent infringement under the Patent Laws of the United States, Title 35 United States Code ("U.S.C.") to prevent and enjoin Defendant Oracle Corporation (hereinafter "Defendant"), from infringing and profiting, in an illegal and unauthorized manner, and without authorization and/or consent from Plaintiff from U.S. Patent No. 7,502,791 ("the '791 Patent" or the "Patent-in-Suit"), which is attached hereto as Exhibit A and incorporated herein by reference, and pursuant to 35 U.S.C. §271, and to recover damages, attorney's fees, and costs.

THE PARTIES

2.   Plaintiff is a Norway company with its principal place of business at Sundlia 16, 1397 Nesoya, Norway.

3.   Upon information and belief, Defendant is a corporation organized under the laws of Delaware, having at least one physical place of business in this judicial district located at 3333 Richmond Road, Suite 420, Beachwood, Ohio 44122. Upon information and belief, Defendant

1

may be served with process c/o The Corporation Trust Company, Corporation Trust Center, 1209

Orange Street, Wilmington, Delaware 19801.

JURISDICTION AND VENUE

4.   This is an action for patent infringement in violation of the Patent Act of the United

States, 35 U.S.C. §§1 *et seq*.

5.   The Court has subject matter jurisdiction over this action pursuant to 28 U.S.C.

§§1331 and 1338(a).

6.   This Court has personal jurisdiction over Defendant by virtue of its systematic and

continuous contacts with this jurisdiction and its residence in this District, as well as because of

the injury to Plaintiff, and the cause of action Plaintiff has risen in this District, as alleged herein.

7.   Defendant is subject to this Court's specific and general personal jurisdiction

pursuant to its substantial business in this forum, including: (i) at least a portion of the

infringements alleged herein; (ii) regularly doing or soliciting business, engaging in other

persistent courses of conduct, and/or deriving substantial revenue from goods and services

provided to individuals in this forum state and in this judicial District; and (iii) being physically

located in this District.

8.   Venue is proper in this judicial district pursuant to 28 U.S.C. §1400(b) because

Defendant resides in this District under the Supreme Court's opinion in *TC Heartland v. Kraft*

Foods Group Brands LLC, 137 S. Ct. 1514 (2017) through its physical location, and regular and

established place of business in this District.

FACTUAL ALLEGATIONS

9.   On March 10, 2009, the United States Patent and Trademark Office ("USPTO") duly and legally issued the '791 Patent, entitled "DATABASE CONSTRAINT ENFORCER" after a full and fair examination. The '791 Patent is attached hereto as Exhibit A and incorporated herein as if fully rewritten.

10.   Plaintiff is presently the owner of the '791 Patent, having received all right, title and interest in and to the '791 Patent from the previous assignee of record. Plaintiff possesses all rights of recovery under the '791 Patent, including the exclusive right to recover for past infringement.

11.   To the extent required, Plaintiff has complied with all marking requirements under 35 U.S.C. § 287.

12.   An exemplary embodiment of a feature claimed in the '791 Patent pertains to a method for enforcing a set of constraints that governs the integrity of information stored in a database system, the constraints being stored in a conceptual rules module in the form of rules for prescribing permitted states and transitions that the database can undertake.

13.   Claim 4 of the '791 Patent states:

"4. Method for enforcing a set of constraints that governs the integrity of information stored in a database system, the constraints being stored in a conceptual rules module in the form of rules for prescribing permitted states and transitions that the database can undertake, the method comprising the steps of

delaying constraint checks until the end of a transaction by creating a check stack during the course of the transaction and executing entries on the check stack at the end of the transaction,

by a stack maker module operatively connected to a runtime module in said database system:

receiving data from said runtime module, and

3

creating and updating said check stack, and retrieving constraints from said conceptual rules module,

wherein the check stack contains a list of functions that have to be executed at the end of the transaction, said functions originating from Insert, Delete and Update Data Manipulation Language (DML) operations calling up the stack maker module,

the Insert DML operation calling up the stack maker module leading to an insert process being performed on the check stack,

the insert process involving placing all checks that have to be executed as a result of an occurrence of a table type being inserted and corresponding conceptual rules being identified for the table type being inserted,

the Delete DML operation calling up the stack maker module leading to a delete process being performed on the check stack,

the delete process involving removing previously inserted entries on the check stack for the occurrence to be deleted and placing all checks that have to be executed as a result of a table type being deleted and corresponding conceptual rules being identified for the table type being deleted, and

the Update DML operation calling up the stack maker module leading to said delete process followed by said insert process being performed on the check stack, and

by an enforcer module: receiving check data from the check stack, processing the check data received from the check stack, and providing resulting data to the runtime module,

wherein said constraints are constraints executed within the transaction which allow conceptual rules to be broken during the transaction, but allow the database system to be in a consistent state at the beginning and end of the transaction." See Exhibit A.

4

14.   As identified in the '791 Patent, prior art systems had technological faults. Ex. A at Col 2:7-11.

15.   Namely, consistency is one of the primary design goals of database systems. *Id.* at Col.1:21-22. Consistency means that the information stored in the database obeys certain constraints defined for the database. *Id.* at Col.1:22-23.

16.   A DML statement is a modification, such as a deletion, an insertion or an update (or modification), of a single piece of information in a database. *Id.* at Col.1:24-26. A transaction is a sequence of DML statements that performs a single logical action in a database application. *Id.* at Col.1:27-28.

17.   One requirement that is deducted from the 100% principle is that all updates storing, deleting or modifying information has to be interrupted and checked by a constraint enforcer. *Id.* at Col.1:34-36. Constraints are a special case of the term "conceptual rules". Conceptual rules are the rules that prescribe all permitted states and transitions a database can undertake. *Id.* at Col.1:36-38.

18.   There are two types of Conceptual Rules, rules of static nature and of dynamic nature. Static rules can be checked at any time, while dynamic rules must be checked for each update. *Id.* at Col.1:44-47.

19.   Some constraints are impossible to implement if they have to be checked per DML statement. *Id.* at Col.1:54-55. One example is the equal constraint. An equal constraint is a rule that says that for a given value in Table T**1**, the same value must exist in Table T**2**, and vice versa. If you insert T**1** first, the value does not exist in T**2** and the insert is rejected. If you insert T**2** first,

the value does not exist in **T1** and the insert is rejected. *Id.* at Col.1:55-60. It is a deadlock situation. *Id.* at Col.1:60-61.

20.   For these kinds of problems, the term Conceptual Transaction has been introduced. *Id.* at Col.1:62-63. It states that at the beginning and end of the transaction, the database must be in a consistent state. *Id.* at Col.1:63-64. During the transaction the database is allowed to be in an inconsistent state. *Id.* at Col.1:65-67.

21.   A Database Transaction is a sequence of DML statements needed for a program to do a certain task. It may be thought of as an envelope with DML statements. *Id.* at Col.2:1-3.

22.   If during the course of a transaction, the Conceptual Rules may be broken, the transaction is referred to as a Conceptual Transaction. *Id.* at Col.2:4-6.

23.   Prior to the '791 Patent, it had been observed that it would be sufficient to check all involved Constraints in a Conceptual Transaction for the total database at the end of the transaction. *Id.* at Col.2:7-9. *But it was also understood that such an approach would be too time consuming for a practical implementation*. *Id.* at Col.2:9-11 (emphasis added).

24.   To address this specific computer-centric problem relating to the processing time, embodiments of the '791 Patent provide, inter alia, (i) a transaction based constraint enforcer for a database system, (ii) a method for enforcing a set of constraints that governs the integrity of information stored in a database, and (iii) a database system, which provides a full constraint check facility, satisfying the 100% principle for databases.

25.   Another way the '791 Patent specifically address and overcomes the computer-centric problems of the prior at was to provide a transaction based constraint enforcer, a method, and a database system, wherein the number of tests that need to be performed at the end of a series of DML statements included in a conceptual transaction does not exceed the number of tests that

6

would have to be performed if the DML statements were not bracketed in a conceptual transaction. *Id.* at Col.2:21-27.

26.   Yet another way the '791 Patent specifically address and overcomes the computer-centric problems of the prior at was to provide a constraint enforcer, a method, and a database system, which includes a transaction based constraint enforcer, wherein conceptual transaction may be implemented in a fashion that allows single DML statements as well as a transaction comprising a sequence of DML statements.

27.   To address this specific technical problem, the '791 Patent's summary provides a method for enforcing a set of constraints that governs the integrity of information stored in a database system is provided. The method comprises the steps of enforcing a set of constraints that governs the integrity of information stored in the database, and delaying constraint checks until the end of a transaction by creating a check stack during the course of the transaction and executing entries on the check stack at the end of the transaction. *Id.* at Col.3:1-6.

28.   Claim 4 of the '791 Patent is a practical application and inventive step of technology that addresses the specific computer-centric problem of time consumption for Constraints in a Conceptual Transaction.

29.   Specifically, to deal with these computer-centric problems, the method of Claim 4 in the '791 Patent requires, inter alia, (a) delaying constraint checks until the end of a transaction by creating a check stack during the course of the transaction and executing entries on the check stack at the end of the transaction, by a stack maker module operatively connected to a runtime module in said database system, (b) by a stack maker module operatively connected to a runtime module in said database system: receiving data from said runtime module, and creating and updating said check stack, and retrieving constraints from said conceptual rules module, and (c)

by an enforcer module: receiving check data from the check stack, processing the check data received from the check stack, and providing resulting data to the runtime module, wherein said constraints are constraints executed within the transaction which allow conceptual rules to be broken during the transaction, but allow the database system to be in a consistent state at the beginning and end of the transaction.

30. These specific elements, as combined, accomplish the desired result of increasing time efficiency for enforcing a set of constraints that governs the integrity of information stored in a database system. Further, these specific elements also accomplish these desired results to overcome the then existing problems in the relevant field of network communication systems. *Ancora Technologies, Inc. v. HTC America, Inc.*, 908 F.3d 1343, 1348 (Fed. Cir. 2018) (holding that improving computer security can be a non-abstract computer-functionality improvement if done by a specific technique that departs from earlier approaches to solve a specific computer problem). See also *Data Engine Techs. LLC v. Google LLC*, 906 F.3d 999 (Fed. Cir. 2018); *Core Wireless Licensing v. LG Elecs., Inc.*, 880 F.3d 1356 (Fed. Cir. 2018); *Finjan, Inc. v. Blue Coat Sys., Inc.*, 879 F.3d 1299 (Fed. Cir. 2018); *Uniloc USA, Inc. v. LG Electronics USA, Inc.*, 957 F.3d 1303 (Fed. Cir. April 30, 2020)

31. Claims need not articulate the advantages of the claimed combinations to be eligible. *Uniloc USA, Inc. v. LG Elecs. USA, Inc.*, 957 F.3d 1303, 1309 (Fed. Cir. 2020)

32. These specific elements of Claim 4 of the '791 Patent were an unconventional arrangement of elements because the prior art methodologies would simply use check all involved Constraints in a Conceptual Transaction for the total database at the end of the transaction. By adding the specific elements of Claim 4 of the '791 Patent was able to unconventionally generate

a method for enforcing a set of constraints that governs the integrity of information stored in a database system. *Cellspin Soft, Inc. v. FitBit, Inc.*, 927 F.3d 1306 (Fed. Cir. 2019)

33.   Further, regarding the specific non-conventional and non-generic arrangements of known, conventional pieces to overcome an existing problem, the method of Claim 4 in the '791 Patent provides a method for enforcing a set of constraints that governs the integrity of information stored in a database system that would not preempt all ways of enforcing constraints because the enforcement is based on, *inter alia*, (a) delaying constraint checks until the end of a transaction by creating a check stack during the course of the transaction and executing entries on the check stack at the end of the transaction, by a stack maker module operatively connected to a runtime module in said database system, (b) by a stack maker module operatively connected to a runtime module in said database system: receiving data from said runtime module, and creating and updating said check stack, and retrieving constraints from said conceptual rules module, and (c) by an enforcer module: receiving check data from the check stack, processing the check data received from the check stack, any of which could be removed or performed differently to permit a method of constraint enforcement in a different way. *Bascom Global Internet Servs., Inc. v. AT&T Mobility LLC*, 827 F.3d 1341 (Fed. Cir. 2016); See also *DDR Holdings, LLC v. Hotels.com, L.P.*, 773 F.3d 1245 (Fed. Cir. 2014)

34.   Based on the allegations, it must be accepted as true at this stage, that Claim 4 of the '791 Patent recites a specific, plausibly inventive way of enforcing a set of constraints that governs the integrity of information stored in a database system. *Cellspin Soft, Inc. v. Fitbit, Inc.*, 927 F.3d 1306, 1319 (Fed. Cir. 2019), *cert. denied sub nom. Garmin USA, Inc. v. Cellspin Soft, Inc.*, 140 S. Ct. 907, 205 L. Ed. 2d 459 (2020).

35. Alternatively, there is at least a question of fact that must survive the pleading stage as to whether these specific elements of Claim 4 of the '791 Patent (i.e., (a) delaying constraint checks until the end of a transaction by creating a check stack during the course of the transaction and executing entries on the check stack at the end of the transaction, by a stack maker module operatively connected to a runtime module in said database system, (b) by a stack maker module operatively connected to a runtime module in said database system: receiving data from said runtime module, and creating and updating said check stack, and retrieving constraints from said conceptual rules module, and (c) by an enforcer module: receiving check data from the check stack, processing the check data received from the check stack) were an unconventional arrangement of elements. *Aatrix Software, Inc. v. Green Shades Software, Inc.*, 882 F.3d 1121 (Fed. Cir. 2018) See also *Berkheimer v. HP Inc.*, 881 F.3d 1360 (Fed. Cir. 2018), *cert. denied*, 140 S. Ct. 911, 205 L. Ed. 2d 454 (2020).

36. Defendant commercializes, inter alia, methods that perform all the steps recited in at least one claim of the '791 Patent. More particularly, Defendant commercializes, inter alia, methods that perform all the steps recited in Claim 4 of the '791 Patent. Specifically, Defendant makes, uses, sells, offers for sale, or imports a method that encompasses that which is covered by Claim 4 of the '791 Patent.

DEFENDANT'S PRODUCT(S)

37. Defendant offers solutions, such as the "Oracle® Fusion Middleware – Oracle Data Integrator" system (the "Accused Product"), that enables a method for enforcing a set of constraints that governs the integrity of information stored in a database system, the constraints being stored in a conceptual rules module in the form of rules for prescribing permitted states and transitions that the database can undertake. A non-limiting and exemplary claim chart comparing

10

the Accused Product of Claim 4 of the '791 Patent is attached hereto as Exhibit B and is incorporated herein as if fully rewritten.

38.   As recited in one step of Claim 4, the system, at least in internal testing and usage, utilized by the Accused Product practices delaying constraint checks until the end of a transaction by creating a check stack during the course of the transaction and executing entries on the check stack at the end of the transaction. See Ex. B.

39.   As recited in another step of Claim 4, the system, at least in internal testing and usage, utilized by the Accused Product practices by a stack maker module operatively connected to a runtime module in said database system: receiving data from said runtime module, and creating and updating said check stack, and retrieving constraints from said conceptual rules module. See Ex. B.

40.   As recited in another step of Claim 4, the system, at least in internal testing and usage, utilized by the Accused Product practices wherein the check stack contains a list of functions that have to be executed at the end of the transaction, said functions originating from Insert, Delete and Update Data Manipulation Language (DML) operations calling up the stack maker module. See Ex. B.

41.   As recited in another step of Claim 4, the system, at least in internal testing and usage, utilized by the Accused Product practices the Insert DML operation calling up the stack maker module leading to an insert process being performed on the check stack. See Ex. B.

42.   As recited in another step of Claim 4, the system, at least in internal testing and usage, utilized by the Accused Product practices the insert process involving placing all checks that have to be executed as a result of an occurrence of a table type being inserted and corresponding conceptual rules being identified for the table type being inserted. See Ex. B.

11

43.   As recited in another step of Claim 4, the system, at least in internal testing and usage, utilized by the Accused Product practices the Delete DML operation calling up the stack maker module leading to a delete process being performed on the check stack. See Ex. B.

44.   As recited in another step of Claim 4, the system, at least in internal testing and usage, utilized by the Accused Product practices the delete process involving removing previously inserted entries on the check stack for the occurrence to be deleted and placing all checks that have to be executed as a result of a table type being deleted and corresponding conceptual rules being identified for the table type being deleted. See Ex. B.

45.   As recited in another step of Claim 4, the system, at least in internal testing and usage, utilized by the Accused Product practices the Update DML operation calling up the stack maker module leading to said delete process followed by said insert process being performed on the check stack. See Ex. B.

46.   As recited in another step of Claim 4, the system, at least in internal testing and usage, utilized by the Accused Product practices by an enforcer module: receiving check data from the check stack, processing the check data received from the check stack, and providing resulting data to the runtime module. See Ex. B.

47.   As recited in another step of Claim 4, the system, at least in internal testing and usage, utilized by the Accused Product practices wherein said constraints are constraints executed within the transaction which allow conceptual rules to be broken during the transaction, but allow the database system to be in a consistent state at the beginning and end of the transaction. See Ex. B.

48.   The elements described in the preceding paragraphs are covered by at least Claim 4 of the '791 Patent. Thus, Defendant's use of the Accused Product is enabled by the method described in the '791 Patent.

INFRINGEMENT OF THE PATENT-IN-SUIT

49.   Plaintiff realleges and incorporates by reference all of the allegations set forth in the preceding paragraphs

50.   In violation of 35 U.S.C. § 271, Defendant is now, and has been directly infringing the '791Patent.

51.   Defendant has had knowledge of infringement of the '791 Patent at least as of the service of the present Complaint.

52.   Defendant has directly infringed and continues to directly infringe at least one claim of the '791Patent by using, at least through internal testing or otherwise, the Accused Product without authority in the United States, and will continue to do so unless enjoined by this Court. As a direct and proximate result of Defendant's direct infringement of the '791 Patent, Plaintiff has been and continues to be damaged.

53.   Defendant has contributed to or induced others to infringe the '791 Patent by encouraging infringement, knowing that the acts Defendant induced constituted patent infringement, and its encouraging acts actually resulted in direct patent infringement.

54.   By engaging in the conduct described herein, Defendant has injured Plaintiff and is thus liable for infringement of the '791 Patent, pursuant to 35 U.S.C. § 271.

55.   Defendant has committed these acts of infringement without license or authorization.

56.   As a result of Defendant's infringement of the '791 Patent, Plaintiff has suffered monetary damages and is entitled to a monetary judgment in an amount adequate to compensate for Defendant's past infringement, together with interests and costs.

57.   Plaintiff will continue to suffer damages in the future unless Defendant's infringing activities are enjoined by this Court. As such, Plaintiff is entitled to compensation for any continuing and/or future infringement up until the date that Defendant is finally and permanently enjoined from further infringement.

58.   Plaintiff reserves the right to modify its infringement theories as discovery progresses in this case; it shall not be estopped for infringement contention or claim construction purposes by the claim charts that it provides with this Complaint. The claim chart depicted in Exhibit B is intended to satisfy the notice requirements of Rule 8(a)(2) of the Federal Rule of Civil Procedure and does not represent Plaintiff's preliminary or final infringement contentions or preliminary or final claim construction positions.

DEMAND FOR JURY TRIAL

59.   Plaintiff demands a trial by jury of any and all causes of action.

PRAYER FOR RELIEF

WHEREFORE, Plaintiff prays for the following relief:

a.   That Defendant be adjudged to have directly infringed the '791 Patent either literally or under the doctrine of equivalents;

b.   An accounting of all infringing sales and damages including, but not limited to, those sales and damages not presented at trial;

c.   That Defendant, its officers, directors, agents, servants, employees, attorneys, affiliates, divisions, branches, parents, and those persons in active concert or participation with any of them, be permanently restrained and enjoined from directly infringing the '791 Patent;

d.   An award of damages pursuant to 35 U.S.C. §284 sufficient to compensate Plaintiff for the Defendant's past infringement and any continuing or future infringement up until the date that Defendant is finally and permanently enjoined from further infringement, including compensatory damages;

e.   An assessment of pre-judgment and post-judgment interest and costs against Defendant, together with an award of such interest and costs, in accordance with 35 U.S.C. §284;

f.   That Defendant be directed to pay enhanced damages, including Plaintiff's attorneys' fees incurred in connection with this lawsuit pursuant to 35 U.S.C. §285; and

g.   That Plaintiff be granted such other and further relief as this Court may deem just and proper.

Dated: September 10, 2020   Respectfully submitted,

                                             SAND, SEBOLT & WERNOW CO., LPA

                                             /s/ Howard L. Wernow
                                             Howard L. Wernow
                                             Aegis Tower - Suite 1100
                                             4940 Munson Street, N. W.
                                             Canton, Ohio 44718
                                             Phone: 330-244-1174
                                             Fax: 330-244-1173
                                             Howard.Wernow@sswip.com

                                             ATTORNEYS FOR PLAINTIFF